

Widget Extras

Focus Area: Front-end



Group Workshop

Back-end

Services
IntegrationContent
ServicesAdvanced
Security

Targeting

Publishing

Front-end

Widget
Development

Widget Extras

Container
Development

Foundation

Portal Essentials

Portal Technologies

Portal Tools

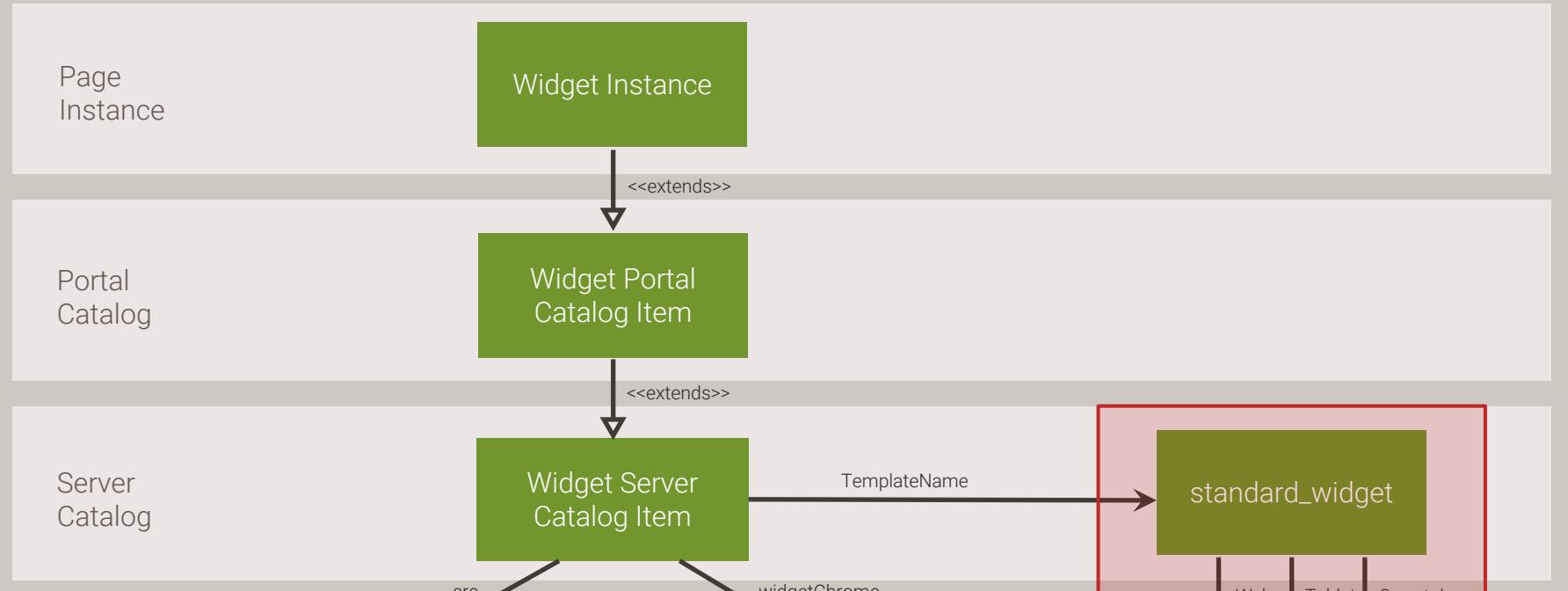
Portal APIs

1. Understanding the Widget Anatomy
2. Customizing ICE Templates for Content Widget
3. Managing Content with Structured Content

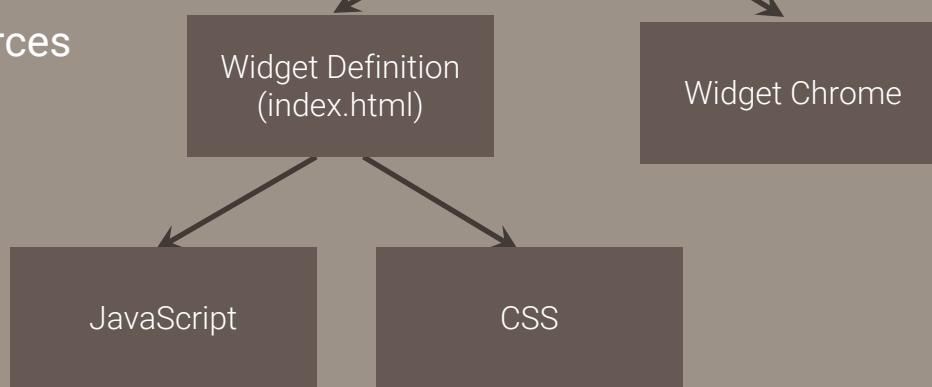
Widget Anatomy

Understanding Widgets

Model



Resources



Standard Widget

Widget Chrome

Widget Chrome Template

```
<div data-pid="{{widgetInstanceId}}>  
  <div class="bp-widget-head"></div>  
  
  <div class="bp-widget-pref"></div>  
  
  <div class="bp-widget-body">{{widgetContent}}</div>  
  
  <div class="bp-widget-foot"></div>  
  
</div>
```

Standard Widget

Widget Chrome

Widget Definition

HTML5
Document

XML Compliant

g:namespace

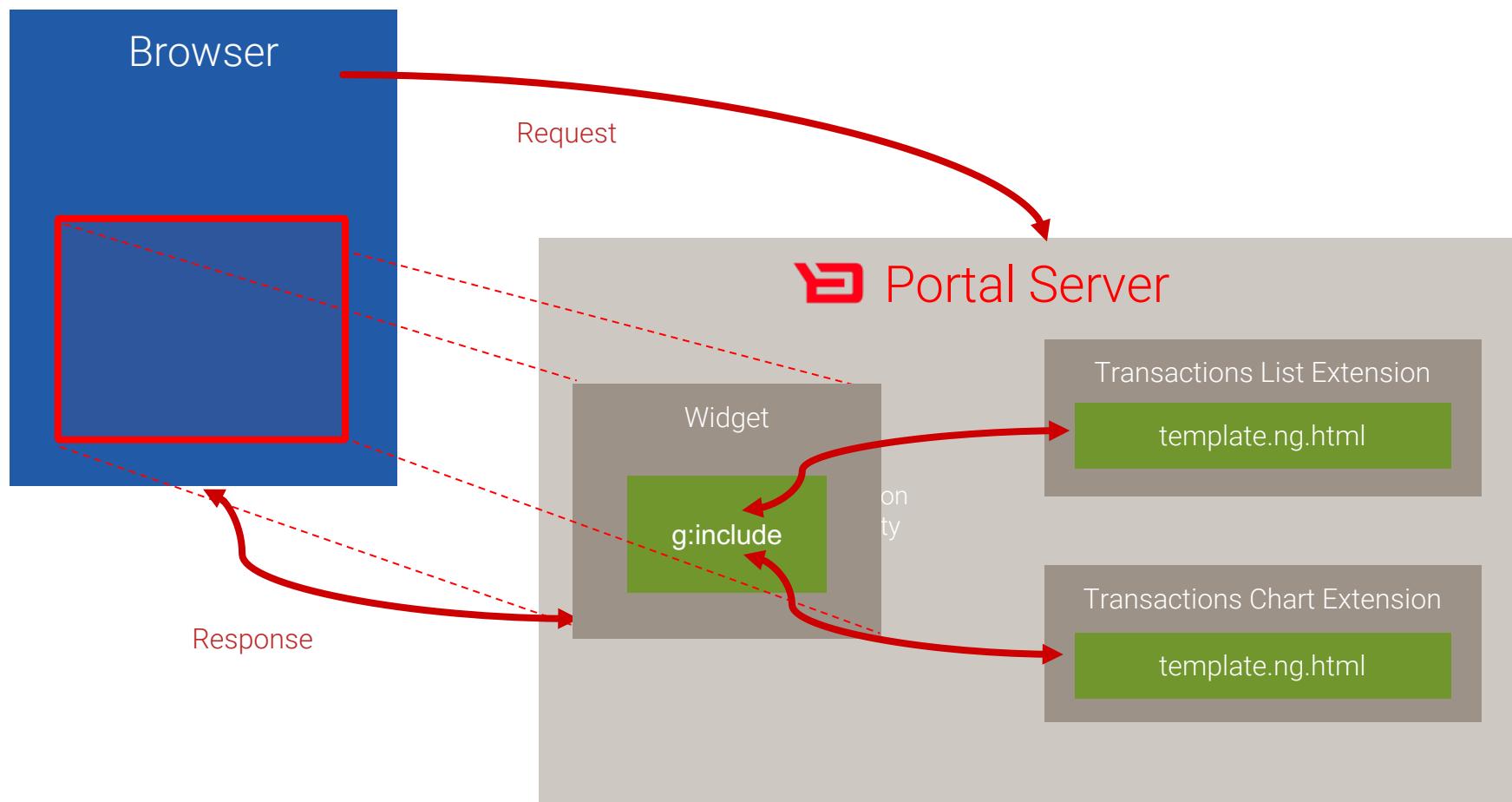
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:g="http://www.backbase.com/2008/gadget" xml:lang="en">
<head>
  <meta charset="utf-8" />
  <title>Transaction List</title>
  <script src="(...)/scripts/vendor-bb-systemjs.js"></script>
  <script src="(...)/scripts/config-bb-module-loader.js "></script>
  <script src="(...)/scripts/lib-bb-start.js"></script>
</head>

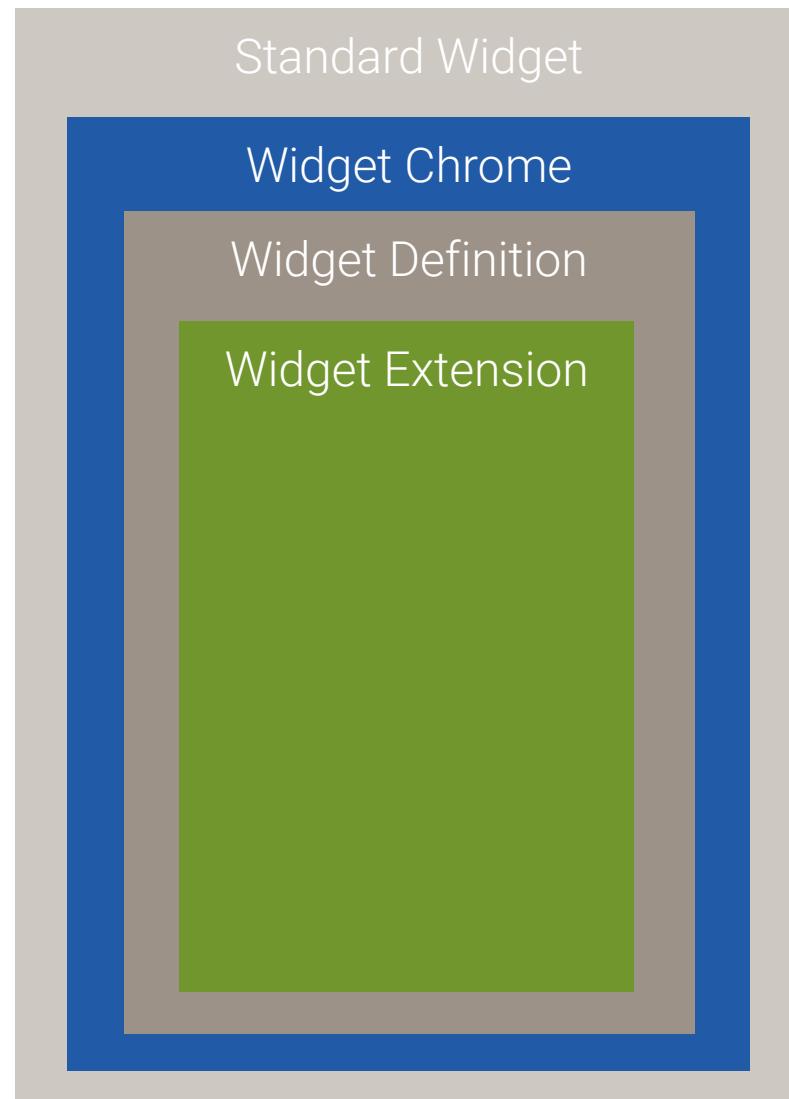
<body g:onload="BB.Widget.ngStart(__WIDGET__);">
  <div ng-cloak="ng-cloak" class="widget-bb-transaction-list-ng">
    <g:include src="${contextRoot}/(...)/${extension}/templates/template.ng.html"
               local="true">
      </g:include>
  </div>
</body>
</html>
```

g:include

g:onload

Rendering using g:include





```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:g="http://www.backbase.com/2008/gadget" xml:lang="en">
<head>
    <meta charset="utf-8" />
    <title>Transaction List</title>
    <script src="(...)/scripts/vendor-bb-systemjs.js"></script>
    <script src="(...)/scripts/config-bb-module-loader.js "></script>
    <script src="(...)/scripts/lib-bb-start.js"></script>
</head>

<body g:onload="BB.Widget.ngStart(__WIDGET__);">
    <div ng-cloak="ng-cloak" class="widget-bb-transaction-list-ng">
        <g:include src="${contextRoot}/(...)/${extension}/templates/template.ng.html"
                  local="true">
        </g:include>
    </div>
</body>
</html>
```

Bank Portal

```
<head>
  <title>Bank Portal</title>
  <script src="vendor-bb-systemjs/scripts/vendor-bb-systemjs.js"/>
  <script src="config-bb-module-loader/scripts/config-bb-module-loader.js"></script>
  <script src="lib-bb-start/scripts/lib-bb-start.js"></script>
</head>
<body>
```

My Stocks Widget

```
<head>
  <script src="vendor-bb-systemjs/scripts/vendor-bb-systemjs.js"></script>
  <script src="config-bb-module-loader/scripts/config-bb-module-loader.js"><script>
  <script src="lib-bb-start/scripts/lib-bb-start.js"><script>
</head>
<body>
```

Account Balance Widget

```
</k
<head>
  <script src="vendor-bb-systemjs/scripts/vendor-bb-systemjs.js"></script>
  <script src="config-bb-module-loader/scripts/config-bb-module-loader.js"><script>
  <script src="lib-bb-start/scripts/lib-bb-start.js"><script>
</head>
<body>

</body>
```

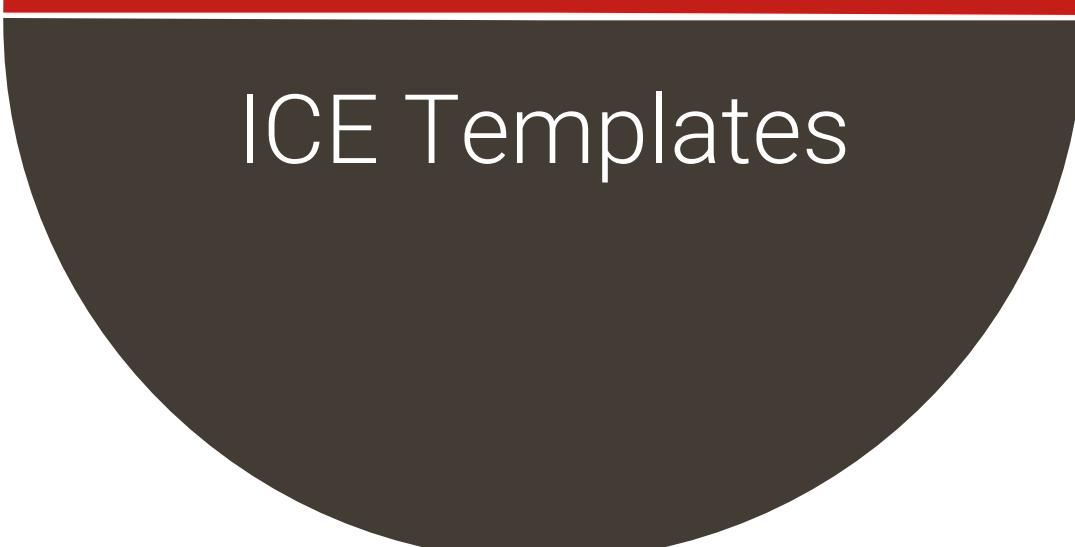
```
</body>
```

Content Widget

Widget Extras



Content Widget



ICE Templates



```
<div class="bt-ice-example">
    <!-- image -->
    

    <!-- metadata -->
    <h2>{{simpleContent.article.bb:title}}</h2>

    <!-- article.content -->
    <div>{{simpleContent.article.content}}</div>

    <!-- Link with attributes -->
    <a href="{{simpleContent.link.path}}"
        title="{{simpleContent.link_desc.bb:title}}">
        {{simpleContent.link_desc.content}}
    </a>

    <!-- Render widget properties -->
    <div title="{{title}}">
        {{title}}
    </div>
</div>
```

Create an ICE Template



Allow the business user to select a new custom content template. Create the template based on the **Twitter Bootstrap Media Object**.

1. Create a custom content template named media-article.html.
2. Import the template as a feature.
3. Make this template available in the templateList property of the content widget.

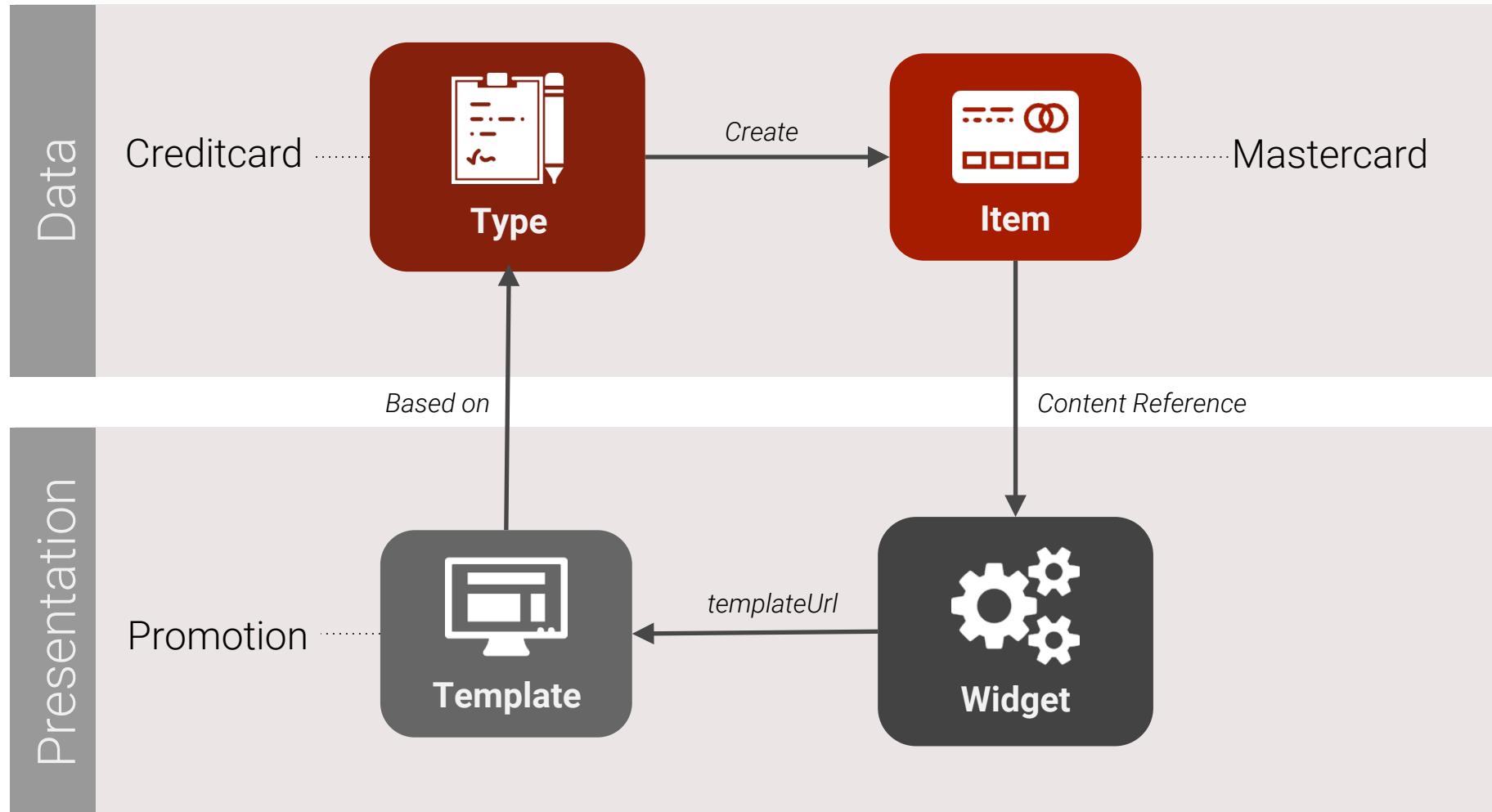
Structured Content

The Basics

*“Structured Content is a feature that offers **reusable content** across portals and control the view through **structured templates**. On top of that also allows you to **manage your content** better through custom Structured Content types.”*

Structured Content

- Structured Content Type
- Structured Content Item
- Structured Content Widget
- Structured Content Template



Structure Content Type

Structured Content Type are created using the JSON Schema Standard.

JSONSchema.net can be used to convert JSON to JSON Schema

Data Validation

Default Values

Type of Inputs

Data Type

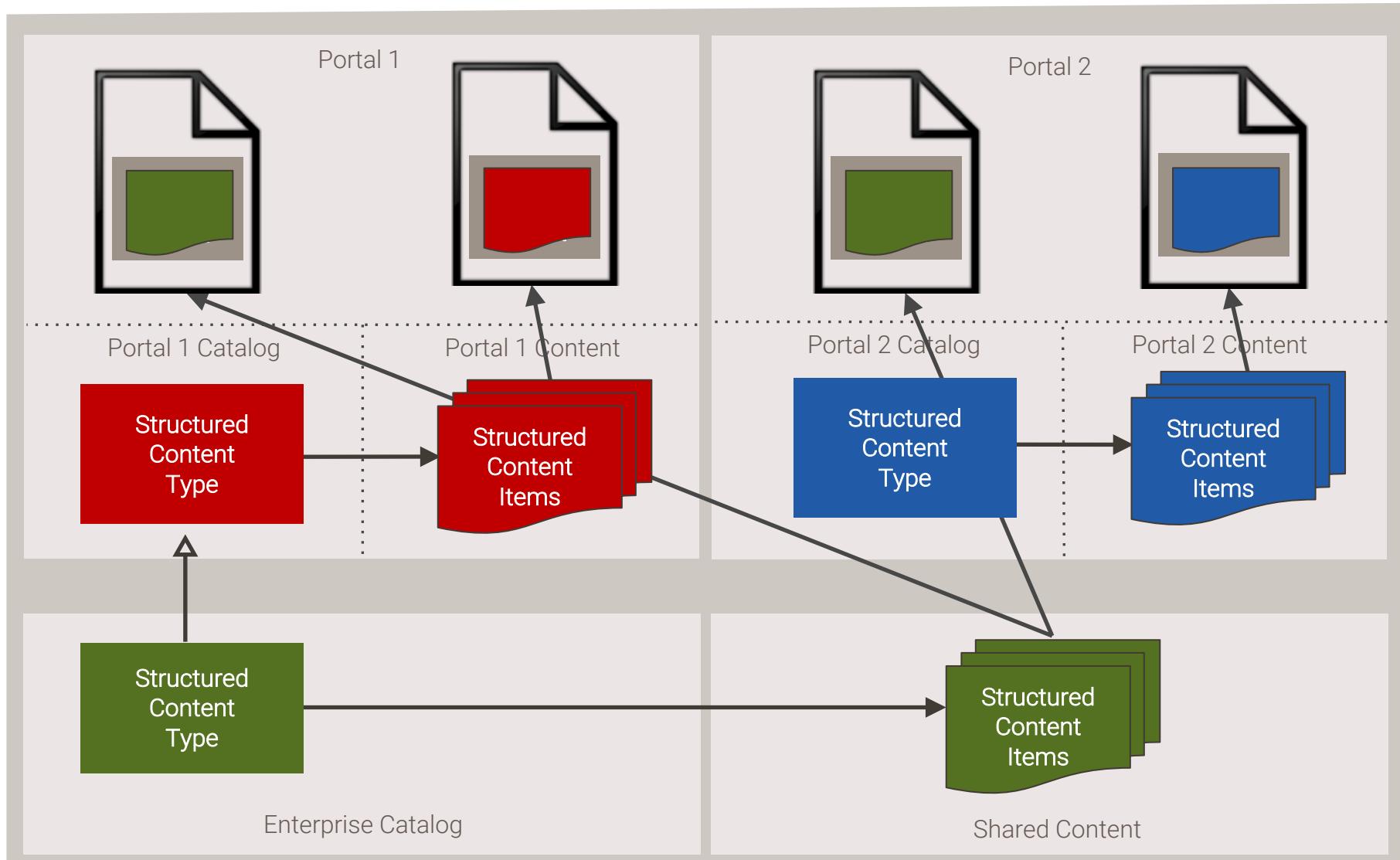
```
{  
    "$schema": "http://json-schema.org/draft-04/schema#",  
    "title": "Buying Reason",  
    "description": "Buying Reason Structured Content.",  
    "type": "object",  
    "properties": {  
        "icon": {  
            "title": "Icon",  
            "type": "string",  
            "metadata": {  
                "infoText": "Drag on an icon image",  
                "renderingHint": "image"  
            }  
        },  
        "title": {  
            "title": "Title",  
            "type": "string",  
            "metadata": {  
                "placeholderText": "Enter the reason title"  
            }  
        },  
        "description": {  
            "title": "Description",  
            "type": "string",  
            "metadata": {  
                "placeholderText": "Describe the essence of this reason"  
            }  
        }  
    }  
}
```

Create a new Structured Content



Allow the business user to create a Structured Content Item based on a Product.

1. Create a new Structured Content Type.
2. Create a new Structured Content Template.
3. Make use of the new Structured Content in your portal.



Thank you!

academy@backbase.com

www.backbase.com